Received: 28 June 2023 | Revised: 5 August 2023 | Accepted: 8 September 2023

IET Cyber-Physical Systems: Theory & Applications

DOI: 10.1049/cps2.12080

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Multiple dependence representation of attention graph convolutional network relation extraction model

Zhao Liangfu | Xiong Yujie | Gao Yongbin | Yu Wenjun

School of Electrical and Electronic Engineering, Shanghai University of Engineering Science, Shanghai, China

Correspondence
Xiong Yujie.
Email: xiong@sues.edu.cn

Funding information
"Science and Technology Innovation Action Plan" of Shanghai Science and Technology Commission for social development project, Grant/Award Number: 21DZ1204900; the Science and Technology Innovation 2030 - Major Project of "New Generation Artificial Intelligence" granted by Ministry of Science and Technology, Grant/Award Number: 2020AAA0109300; Shanghai Local Capacity Enhancement project, Grant/Award Number: 21010501500

Abstract
Dependency analysis can better help neural network to capture semantic features in sentences, so as to extract entity relation. Currently, hard pruning strategies and soft pruning strategies based on dependency tree structure coding have been proposed to balance beneficial additional information and adverse interference in extraction tasks. A new model based on graph convolutional networks, which uses a variety of representations describing dependency trees from different perspectives and combining these representations to obtain a better sentence representation for relation classification is proposed. A newly defined module is added, and this module uses the attention mechanism to capture deeper semantic features from the context representation as the global semantic features of the input text, thus helping the model to capture deeper semantic information at the sentence level for relational extraction tasks. In order to get more information about a given entity pair from the input sentence, the authors also model implicit co-references (references) to entities. This model can extract semantic features related to the relationship between entities from sentences to the maximum extent. The results show that the model in this paper achieves good results on SemEval2010-Task8 and KBP37 datasets.

KEYWORDS
complex networks, data analysis, information networks

1 | INTRODUCTION

Entity relationship extraction is a subordinate task of information extraction. The object of entity relationship extraction task is to extract entity relationship triplet from unstructured text, namely < entity 1, relationship, entity 2>, in which 'entity 1' and 'entity 2' are two named entities involved in 'relationship', and 'relationship' refers to the type of relationship between two entities. Entity relation extraction is the key technology in semantic understanding, and also the basis of machine translation, knowledge graph construction, automatic question answering system and other applications.

At present, there are two main research frameworks for entity relationship extraction: one is pipeline method, that is, entity relationship extraction is carried out after entity recognition. The second is the joint extraction method, that is, entity identification and relation extraction at the same time. The pipeline method carries out relationship extraction on the basis of named entity recognition. Errors generated in entity recognition will affect relationship prediction results and cause error propagation [1]. Compared with the pipelined method, the combined extraction method is considered to have better performance and potential. In 2017, ZHENG et al. [2] earlier proposed an entity relationship joint extraction method based on the new labelling strategy, which converted the joint learning model containing named entity recognition and relationship classification into a sequential labelling problem and achieved good results. Although joint relation extraction avoids the error propagation problem in the pipeline method, it requires more complex model structure to encode richer semantic information.

The purpose of dependency analysis is to reveal the syntactic structure of a sentence by analysing the dependencies among the components. The dependency analysis information representing the syntactic structure of the text grammar can provide effective prior structured text information for the joint relation extraction, help the model to clarify the text structure, and improve the entity relation extraction performance. In ref. [3], dependency analysis and heuristic rules of Chinese grammar are used to extract relational statements, then argument positions are determined according to distance, and triples are finally output, thus avoiding the restrictions on relational extraction brought by complex Chinese grammatical rules, flexible expression methods, and diversified semantics. In ref. [4], word sequences based on the shortest dependency path (SDP) are added to the model input, and semantic information of the text is extracted through Bidirectional Long Short Term Memory (Bi-LSTM) networks and convolutional neural networks. It has achieved good results in Chinese news corpus. Dependency analysis builds a syntactic tree structure. Considering the complexity of Chinese syntactic structure, the graph method is introduced to encode the structural information in dependency analysis, which has higher flexibility and applicability than the traditional tree structure. Graph Convolutional neural Network (GCN) is an implementation of convolutional networks, which can extract the spatial features in the topology diagram and effectively aggregate the entity nodes containing entity relationships, thus improving the performance of entity relationship extraction. In order to reduce information redundancy, researchers cut the dependency relationships in the dependency analysis diagram and only retain part of the dependency relationships [5–8].

Based on the above discussion, this paper explores a more comprehensive and flexible way to enhance dependency structure coding. We present a RE model based on GCN highlighting Multiple dependency representations (abbreviated as MDR-GCN for the rest of this paper). In the model presented in this paper, the GCN sublayer is driven in parallel by three dependency representations, including the full adjacency matrix, the centralised adjacency matrix and the distance-weighted adjacency matrix. These matrix representations describe the dependency structure with different refinements. The output of the GCN sublayer is then input into the GSF Extractor module. This module uses the attention mechanism to capture deeper semantic features from the context representation as the global semantic features of the input text, thus helping the model to capture deeper semantic information at the sentence level for relational extraction tasks. In order to get more information about a given entity pair from the input sentence, this article also models implicit co-references (references) to entities. This operation fully and flexibly encodes the entire dependency structure and preserves the most useful dependency information needed for relationship classification. In addition, the renormalisation parameter is introduced into the graph convolution operation, which will also affect the accuracy of the whole RE model. Therefore, this parameter is constantly adjusted in the training process to obtain the best performance of the proposed RE model. The model was trained and tested on Semeval2010Task8 and KBP37 datasets.

In addition, a large amount of data has been generated in the industrial digital age, and how to transform the data into valuable knowledge is worth studying. Fault diagnosis and root cause analysis of industrial equipment play an important role in the whole process of automatic production. Because the structure of industrial large equipment is very complex, once the fault occurs, it may affect the whole production cycle. It is very important to accurately diagnose the fault of industrial equipment and make it in the best working state. By exploring the complex correlation between faults, knowledge analysis and utilisation can be realised, and auxiliary decision making and diagnostic reasoning can be provided for maintenance personnel, which is of great research significance. The model proposed in this paper has also achieved good results on the data set of industrial equipment fault diagnosis and root cause analysis.

2 | RELATED WORK

Traditional relational extraction mainly constructs classification models based on features [9] or kernel functions [10]. This method is feasible and effective, but depending on selected feature sets or designed kernel functions, it is easy to introduce human errors, which limits the performance of relational extraction models to a large extent.

At present, deep learning-based methods are widely used in relation extraction tasks. The authors in refs. [11] and [12] extract sentence sequence features using CNN and RNN respectively, and implement relational classification through Softmax classifier. In view of the noise problem caused by data imbalance, Santos et al. [13] proposed a sort loss function to replace cross entropy, and carried out special processing on other classes to reduce the influence of noise. Wang et al. [14] introduce the attention mechanism into the relational extraction model and focuses on the effective information in sentences through the attention mechanism to improve the performance of the model. Considering that local features and context features of sentences contribute to the task of relation extraction, the authors in refs. [15] and [16] used the method of joint neural network to combine RNN and CNN to obtain local features and context features of sentences, so as to improve the performance of model relation extraction. In the above model, the original statement is directly used as the input to construct the end-to-end model, and good results are obtained.

In addition, in order to fully explore the deep semantic information in sentences, the researchers import the dependency tree of sentences into the model and build a model based on dependency relationship. In order to make full use of the effective information in dependency tree and eliminate the interference features, the researchers put forward a variety of pruning strategies to select the beneficial information in dependency tree. Xu et al. [17] popularised the idea of Dependency based on SDP between entities by pruning and applied it to LSTM networks. Guo et al. [18] have used a Recurrent Convolutional Neural Network model in which an attention mechanism based on the SDP is added to enhance keywords and sentence features. In ref. [19], pruning strategy is applied to reduce the entire tree to the subtree under the entity Lowest Common Ancestor (LCA), and the structural information of the subtree is captured by LSTM-RNN with bidirectional tree structure. Zhang et al. [5] proposed an improvement on the basis of LCA rules, preserving nodes within K distance of LCA subtree of entity pairs, and introducing graph convolutional network for relation extraction. The above research shows that dependency trees contain abundant information that is beneficial to the task of relation extraction, which plays a certain role in improving the performance of relation extraction model. However, the rule-based hard pruning strategy tends to lead to over-pruning or under-pruning, thus reducing the utilisation rate of information in dependency trees. Moreover, most models choose CNN or RNN as feature extractors. Non-local dependency features in dependency tree cannot be fully learned.

3 | MODEL DESCRIPTION AND IMPLEMENTATION

Model Description and implementation. This chapter mainly introduces the graph convolutional neural network based on multiple dependency representation. The model framework is shown in Figure 1. The structure and function of each module will be described in detail below.

3.1 | Introduction to the overall framework of the model

The model in this paper is inspired by ref. [20], pretraining model BERT is used to encode input sentences, and then a graph convolution neural network model based on multiple

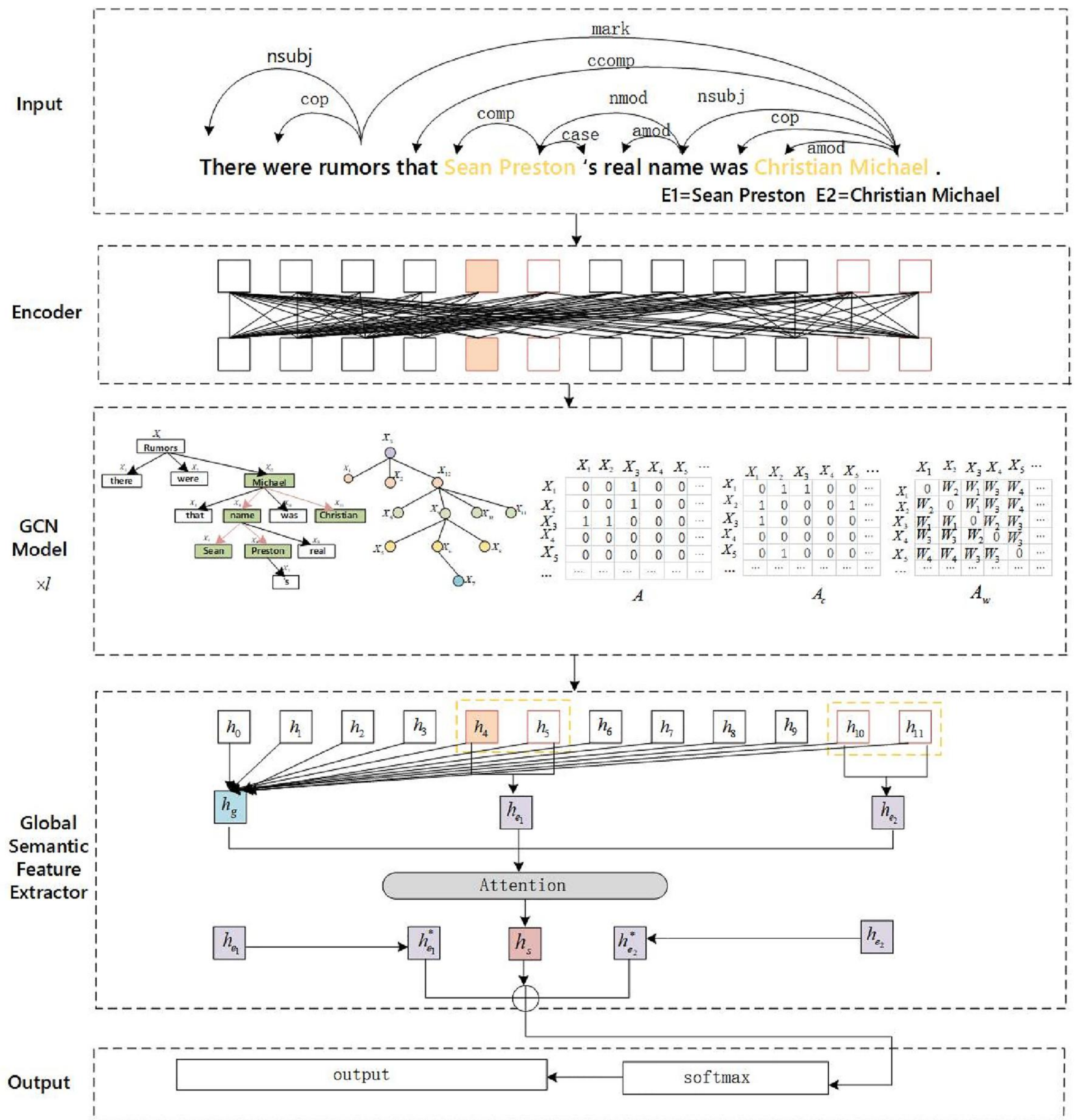

FIGURE 1 Graph neural network model framework for multiple dependence representation.

dependency representation is used. Specifically, the model uses multiple representations to describe the dependency tree from different perspectives, and then combines these representations to obtain a better sentence representation for relational classification. This model can extract the semantic features related to the relationship between entities from sentences to the maximum extent. The output is input into the new GSF Extractor module. The increased GSF Extractor module uses the attention mechanism to capture deeper semantic features from the context representation as the global semantic features of the input text, which can help the model to capture deeper semantic information at the sentence level for relational extraction tasks. Finally, the output results are forecasted by softmax operation.

3.2 | BERT model

The BERT pre-training model used in this paper is essentially a multi-head attention mechanism model built with the encoder of Transformer model as the base model. In previous studies, BERT and RNNs models are different, and their performance in parallelism and semantic understanding is more prominent.

The BERT model is just an encoder using the Transformer model, which is made up of 6 identical layers, each of which is made up of two sub-layers. The processed words in the original sequence are mapped to multidimensional word vectors $e_i \in R^d$, d is the dimension of the word vectors. Then, we get the set of word vectors for the sentence s $X = \{e_1,\cdots,e_n\}$, where $X \in R^{n\times d}$. Therefore, the multi-head attention mechanism layer can be roughly represented as follows:

$$z = \mathrm{MultiHead}(Q, K, V) = HW^o \tag{1}$$

where $W^o \in R^{hn\times k}$ is the weight matrix of the bull's attention.

In the model, the multi-head self-attention means that Q, K and V are firstly transformed linearly, and then the similarity is calculated. This process is repeated for h times, and then the results of h times are combined together and then linear transformation is performed as the result of the multi-head self-attention mechanism. The calculation method is as follows:

$$Q = XW^Q \tag{2}$$

$$K = XW^K \tag{3}$$

$$V = XW^V \tag{4}$$

where $W^Q \in R^{k\times n}$, $W^K \in R^{k\times n}$, $W^V \in R^{k\times n}$ are the weight matrices of Q, K and V respectively. Then, after repeating a times, the final output of multi-head attention is to splice the output of each head. The expression is as follows:

$$H = \mathrm{head}_1 \oplus \mathrm{head}_2 \oplus \cdots \oplus \mathrm{head}_h \tag{5}$$

where $H \in R^{n\times hn}$, $\oplus$ is a concatenation operation. To sum up, the expression of $head_i$ is as follows:

$$\mathrm{head}_i = \mathrm{softmax}\left(\frac{\left(XW_i^Q\right)\left(XW_i^K\right)^T}{\sqrt{k}}\right)\left(XW_i^V\right) \tag{6}$$

where $W_i^Q \in R^{k\times n}$, $W_i^K \in R^{k\times n}$, $W_i^V \in R^{k\times n}$. After residual and normalisation processing, the results of multiple attention mechanism layer enter the feedforward neural network layer, which obtains the vector representation of text semantics through simple linear activation operation. The process is as follows:

$$C = \max[0, ZW_1 + b_1]W_2 + b_2 \tag{7}$$

where W_1 and W_2 are the weight matrix of the feedforward network; b_1 and b_2 are the bias of the feedforward network.

3.3 | Dependent propagation layer

3.3.1 | Graph convolutional network

GCN is a simple and effective graph-based convolutional neural network, which can effectively capture the dependency between data through the information transmission between graph nodes. Therefore, it is often used to process the data with rich and interdependent relationships between objects. The input of the graph convolutional network is the structure of the graph and the characteristic representation of the nodes in the graph. For each node in the graph, GCN obtains the feature representation vector of the node through the property fusion of other nodes near the node.

Given the adjacency matrix $A = \left(a_{ij}\right)_{n\times n} \in R^{n\times n}$ and matrix input data $H^{(0)} \in R^{n\times d}$, each layer in the GCN structure of layer l can be represented by the graph convolution operation as follows:

$$\begin{aligned} H^{(m)} &= GCN\left(H^{(m-1)}; A\right) \\ &\triangleq \sigma\left(\tilde{A}H^{(m-1)}W_A^{(m)} + B_A^{(m)}\right), m = 1, 2, \cdots, l \end{aligned} \tag{8}$$

where $H^{(m)}$ is the output of layer m and the input of layer $(m+1)$, $W_A^{(m)} \in R^{d\times d}$ and $B_A^{(m)} \in R^{n\times d}$ are the linear parameter matrix and bias of layer m; $\sigma(.)$ is a non-linear activation function, such as ReLU function. $\tilde{A}H^{(m-1)}W_A^{(m)}$ is the graph convolution operation with respect to A; $\tilde{A}$ is a renormalised version of A. In the GCN operation of the model proposed in this paper, $\tilde{A}$ is defined as follows:

$$\tilde{A} = (D + \gamma I)^{-1}(A + \gamma I) \tag{9}$$

where I is the identity matrix of order n;

The degree matrix $D = (d_{ij})_{n\times n}$ is a diagonal matrix of $(d_{ij})_{n\times n}$, defined as follows:

$$d_{ii} = \sum\nolimits_{j=1}^{n} a_{ij}, d_{ij} = 0(i \neq j) \tag{10}$$

where γ is the renormalisation parameter, which can be adjusted to achieve the best result.

Referring to the description of renormalisation parameters in the paper [20], Kipf and Welling propose the GCN in ref. [21], and from their introduction, it is inspired not only by spectral graph theory based works [22, 23], but also by the graph learning models. They suggest that a renormalisation trick should be introduced to ensure the numerical stability of the graph convolution operator. Inspired by the K-scaling method, we assume that if we introduce a renormalisation parameter γ to the renormalised adjacency matrix in the graph convolution operator, adjust γ in training, then it is possible to obtain a better result than what have been achieved by previous GCN-based RE frameworks, where such renormalisation parameter is not explicitly included and is essentially set to 1 by default. Wu et al. [24] perform a spectral graph theoretical analysis, which is also established for our proposed RE framework.

3.3.2 | Presentation dependency structure

The result of dependency analysis is usually a graph where the nodes represent the tags and the edges represent the dependencies between the tags, called a dependency graph. Given an entity pair, the SDPS in the dependency tree is defined as the shortest path between the respective node subsets corresponding to the two entities. Each SDPS considered in this article are unique paths between the LCA (LCAs) pairs of their respective subset of entity nodes. To encode the dependency structure, you need to transform the dependency tree into a computable mathematical object. This article covers some of the following methods.

1) Full adjacency matrix

An adjacency matrix is a common algebraic representation of the adjacency structure of a graph or tree. The dependency structure can be encoded using the adjacency matrix of the complete dependency tree. The adjacency matrix A of a graph or tree containing n vertices is a matrix $(a_{ij})_{n\times n}$, where if there is an edge connection between nodes i and j, then $a_{ij} = 1$; Otherwise, $a_{ij} = 0$.

However, previous relational extraction work has shown that variants of the adjacency matrix may have better performance than the original adjacency matrix. In the model proposed in this paper, these different adjacency matrices will be combined in an attempt to combine the information provided by these matrices to improve the performance of entity relationship extraction.

2) Concentrated adjacency matrix

Xu et al. [17] proposed an LSTM-based RE framework approach that only utilises SDP instead of the full dependency tree of the original sentence, which achieves better performance than the traditional RNN-based model (in which there is no dependency structure). This enables the use of adjacency matrices centred on SDP in GCN operations. This type of matrix can be obtained by setting the items representing edges not included in SDP as 0 in the given original adjacency matrix A, that is, SDP-concentrated adjacency Matrix A_c is defined by a matrix $(c_{ij})_{n\times n}$: if nodes i and j are connected by an SDP edge, $c_{ij} = 1$; Otherwise, $c_{ij} = 0$. This is equivalent to degrading the full dependency tree to its subgraph, where all nodes on the SDP side are retained, but nodes on the non-SDP side are removed. As a result, SDPS are emphasised, and GCN operations are more focused on SDP-related data.

3) Distanced-weighted adjacency matrix

Contrary to the centralised method, this paper retains all the non-zero terms in the original adjacency matrix and modifies the zero terms according to the distance between nodes, which is defined as the length of the shortest path between these nodes. This is equivalent to converting the original tree into a weighted complete graph whose edge weights are determined by distance. This article accomplishes this by assigning weights to each node pair:

If $i \neq j$, $w_{ij} = \varphi(d_{ij})$; Otherwise, $w_{ij} = 0$.

Where w_{ij} and d_{ij} are the weights and distances of nodes i and j respectively, and $\varphi(.)$ is the non-negative decreasing function of distance. For example, Shuman et al. [25] set φ as threshold Gaussian kernel function, ref. [25] shows that when the GCN layer is calculated with the distance-weighted adjacency matrix, more comprehensive dependency analysis information can be used and the accuracy of relational extraction task can be improved.

3.3.3 | Joint representation of multi-dependency relationships

As mentioned earlier, various types of matrix representations of dependency structures are used in the GCN-based approach. However, as far as is known, there is insufficient evidence or rigorous theoretical analysis to say which dependency representation is optimal for a given GCN-based model or for a specific NLP task. Therefore, as an attempt to present the model in this paper, various types of dependency representations are combined, including the adjacency matrix A, SDP-concentrated adjacency matrix A_c and exponential-distance-weighted adjacency matrix A_w. The combined output of the parallel GCN layer driven by different representations of the dependency tree is then input into the new module GSF Extractor module. The initial input $H_{(.)}^{(0)}$, intermediate input $H_{(.)}^{(m)}$ and the relation extraction $H^{(l)}$ between output and final output of the model proposed in this paper are shown as follows:

$$H_a^{(0)} = H_c^{(0)} = H_w^{(0)} = [h_1 h_2 \ldots h_s]^T \in \Re^{s \times d} \tag{11}$$

$$H_a^{(m)} = GCN\left(H_a^{(m-1)}; A\right), m = 1, 2, \ldots, l \tag{12}$$

$$H_c^{(m)} = GCN\left(H_c^{(m-1)}; A_c\right), m = 1, 2, \ldots, l \tag{13}$$

$$H_w^{(m)} = GCN\left(H_w^{(m-1)}; A_w\right), m = 1, 2, \ldots, l \tag{14}$$

$$H^{(l)} = \left[H_a^{(l)} \quad H_c^{(l)} \quad H_w^{(l)} \right] \in \Re^{s \times 3d} \tag{15}$$

According to the operation mentioned in ref. [5], a sentence representation $f_s : \Re^{s \times 3d} \rightarrow \Re^{3d}$ in vector form can be obtained through a maximum pooling operation h_s, which is defined as follows:

$$h_s = f_s\left(H^{(l)}\right) \tag{16}$$

In this article, $h_i^{(m)}$ is denoted as the output of layer m x_i and the input of layer $(m + 1)x_i$, which is row i of matrix $H^{(m)}$.

3.3.4 | Global semantic feature extractor module

The output of the graph convolutional network can be denoted by the following formula:

$$H = \{h_0, \ldots, h_n\} \tag{17}$$

In this paper, the maximum pooling operation is used to obtain the shallow characteristics of entity pairs and processed input sentences. Entity index pairs of entity e_1 are denoted as (i,j) and entity index pairs of entity e_2 are denoted as (k,l), as shown in formulas (18), (19) and (20).

$$h_{e1} = \text{Maxpooling}\left(h_{i:j}\right) \tag{18}$$

$$h_{e2} = \text{Maxpooling}(h_{k:l}) \tag{19}$$

$$h_g = \text{Maxpooling}(H) \tag{20}$$

Previous work has always directly linked vector representation $[h_{e1} \oplus h_{e2} \oplus h_g]$ as the global semantic feature of the input text. This paper argues that this is not enough to help the model capture deeper sentence-level semantic information for relational extraction tasks. Unlike them, in order to obtain better global sentence-level semantic features, this paper uses a global semantic feature extractor module, which uses $[h_{e1} \oplus h_{e2} \oplus h_g]$ as a query vector to capture deeper semantic features from context representation H, as shown in Equation (21).

$$h_s = \text{Softmax}\left(\frac{H \cdot \left(W_s\left[h_{e1} \oplus h_{e2} \oplus h_g\right]\right)}{\sqrt{d}}\right) \cdot H \tag{21}$$

where $W_s \in R^{d \times 3d}$ is a linear transformation matrix, and d is the hidden dimension of the vector.

To get more information about a given entity pair from the input sentence, the article also models implicit co-references (references) to entities. Specifically, this paper uses the representation of the entity as the query vector to obtain a new entity feature vector from H, as shown in formulas (22) and (23).

$$h_{e1}^* = \text{Softmax}\left(\frac{H \cdot h_{e1}}{\sqrt{d}}\right) \cdot H \tag{22}$$

$$h_{e2}^* = \text{Softmax}\left(\frac{H \cdot h_{e2}}{\sqrt{d}}\right) \cdot H \tag{23}$$

3.3.5 | Relational classification layer

Finally, a trainable matrix W_R is used to map it to the output space:

$$O = W_R \cdot \left(h_s \oplus h_{e1}^* \oplus h_{e2}^*\right) \tag{24}$$

where O is an $|R|$-dimensional vector, and each value represents a relation type in the relation type set R. softmax function is used to predict the relation $\hat{r}$ between e_1 and e_2:

$$\hat{r} = \text{argmax} \frac{\exp(O^u)}{\sum_{u=1}^{|R|} \exp(O^u)} \tag{25}$$

where O^u represents the value of vector O in the u dimension.

4 | EXPERIMENT AND RESULT ANALYSIS

4.1 | Experimental data and evaluation index

This paper conducts experiments on two standard relational extraction data sets, which are:

1) SemEval2010-Task8 data set. The dataset contains 10,717 sentence instances, including 8000 training instances and 2717 test instances. The relationship types include 9 class relations and 1 other class.
2) KBP37 data set. Using the 2013 and 2010 KBP documentation datasets and the 2013 Wikipedia annotated text datasets, this dataset includes 15,917 training instances, 3405 test instances, and 19 different relationships in which low-frequency relationships are discarded, with more than 100 training instances for each relationship. The statistics of the two datasets are shown in Table 1, and the details of the SemEval2010 and KBP37 datasets are shown in Table 2.

The evaluation model of Macro F1 value was adopted in both data sets. Macro first calculates F1 for each class, then computes an arithmetic average for all classes. Before calculating the F1 index value, the precision ratio P and recall ratio

R were obtained according to the confusion matrix, as shown in Equations (26) and (27).

$$P = TP/(TP + FP) \times 100\% \tag{26}$$

$$R = TP/(TP + FN) \times 100\% \tag{27}$$

F1 value is defined as the harmonic average of the precision rate and recall rate, as shown in Equation (28).

$$F_1 = 2PR/(P + R) \tag{28}$$

4.2 | Parameter setting

General training Settings such as hyperparameters follow the description in ref. [5] and are created or adjusted according to the model and experimental sections proposed in this paper. See Table 3 for details.

TABLE 1 The statistics of the dataset.

Dataset	Number of training instances	Number of test instances
Semeval2010 Task8	8000	2717
KBP37	15,917	3405

4.3 | Existing model

Several state-of-the-art RE methods, most of which are dependency based, are used as a baseline to compare the model in this paper:

1) CNN + PF (CNN Position Feature) [11]: The model is basic CNN, introduce entity location characteristics.
2) RNN + PF [12]: Replace CNN in CNN + PF with the basic RNN.
3) Att-Bi-LSTM (Attention Bi-LSTM) [26]: Using the attention mechanism The LSTM output layer is used to capture important semantic features in sentences.
4) SDP-LSTM [17]: Through pruning strategy, the SDP in dependency tree is selected as the input, and LSTM is used to extract heterogeneous information.
5) SPTree (Shortest Path Tree) [18]: The pruning strategy is applied to reduce the whole tree to the subtree under the LCA of the entity, and the LSTM of the bidirectional tree structure is used to capture the higher-order features of sentences.
6) SA-Bi-LSTM-LET [27]: combines entity perception attention mechanism with potential entity type, and makes full use of entity information for relationship extraction.
7) Tree-LSTM [28]: Tai et al. emphasised the advantages of LSTM in representing sequentially sensitive sequences, and designed a Tree-LSTM model to obtain the semantic relevance of dependent Tree information.

TABLE 2 Details about the SemEval2010 and KBP37 datasets.

SemEval2010 Task8		
Number of relation types	19	
Relation category number	10	
Types	Cause-effect	Instrument-agency
	Product-producer	Content-container
	Entity-origin	Entity-destination
	Component-whole	Member-collection
	Communication-topic	Other
KBP37		
Number of relation types	37	
Relation category number	19	
Types	per:alternate_names	org:alternate_names
	per:origin	org:subsidiaries
	per:spouse	org:top_members/employees
	per:title	org:founded
	per:employee_of	org:founded_by
	per:countries_of_residence	org:countries_of_headquarters
	per:stateorprovince_of_residence	org:stateorprovince_of_headquarters
	per:country_of_birth	org:member
	No_relation	

8) PA-LSTM [29]: Zhang et al. adopted the entity location-aware attention mechanism in the LSTM sequence model. This model does not use a dependency tree.
9) C-GCN [5]: Zhang et al introduced graph convolutional network and adopted pruning strategy to remove contents unrelated to SDPS to the maximum extent.
10) C-AGGCN [30]: Guo et al. adopted the multi-head self-attention mechanism to calculate the weight of each node in the dependency tree. To get more structural information, they added dense connections to the model.
11) C-MDR-GCN [20]: The model uses multiple representations to describe the dependency tree from different perspectives and groups these representations together to obtain a better sentence representation for relational classification.

4.4 | Experimental result

To test the performance of the proposed model (MDR-GCN + GSF Extractor), we evaluated it on the SemEval 2010Task8 and KBP37 tasks. The results are shown in Table 4. In both data sets, the value of F1 is higher than that of the existing model. In addition, the GSF Extractor module added to the model in this paper uses the attention mechanism to capture deeper semantic features from the context representation as the global semantic features of the input text, so as to help the model to capture deeper semantic information at the sentence level for relational extraction tasks.

TABLE 3 Experiment parameter.

Hyper-parameter	Value
Dropout rate	0.5
Renormalisation parameter	[0.25, 1.75]
Learning rate	0.5
Batch size	50
GCN layer	2

TABLE 4 F1 score of the improved A-GCN model.

Model	Semeval 2020Task8	KBP37
CNN + PF	82.7%	51.3%
RNN + PF	82.5%	54.3%
Att-Bi-LSTM	84.0%	-
SDP-LSTM	83.7%	58.3%
SPTree	84.4%	59.1%
SA-Bi-LSTM-LET	85.1%	59.0%
Tree-LSTM	84.4%	59.1%
PA-LSTM	82.7%	61.4%
C-GCN	84.8%	63.6%
C-AGGCN	85.1%	64.3%
C-MDR-GCN	84.9%	65.2%
MDR-GCN + GSF extractor	89.4%	69.2%

4.5 | Ablation experiment

To investigate the contribution of each component in the model presented in this paper, ablation experiments were performed on the Semeval 2020 Task8 test dataset (see Table 5). Each component shown in Table 5 is removed, and the corresponding F1 value reflects the contribution of the corresponding component. It shows that all components contribute positively to our model. Among them, the contribution of the dependent propagation layer represented by the full adjacency matrix A, the centralised adjacency matrix A_c and the distance-weighted adjacency matrix A_w is the largest. By contrast, you can combine either or both of these three dependencies and the F1 value will decrease accordingly. The results show that the collaborative work of A, A_c and A_w can significantly improve the overall performance of the dependent propagation module. A_w In the model of this paper, GSF Extractor module is also an important part. In general, the MDR-GCN + GSF Extractor model in this paper shows its effectiveness on RE structures. A_w obviously has a greater impact on the experimental results than A and A_c. We speculate the reason, which may be because the information contained in A_w, such as the shortest path length between nodes, is more important to the experimental results.

4.6 | The effect of sentences of different lengths

Figure 2 compares F1 values of C-GCN, C-AGGCN, C-MDR-GCN and MDR-GCN + GSF Extractor (our model) for different sentence lengths. In the Semeval2010Task8 task, the test set is divided into five levels based on sentence

TABLE 5 The ablation of the model in this paper was performed on the Semeval2010Task8 test set.

Model	F1 score
MDR-GCN + GSF extractor	89.2%
$-A$	87.3%
$-A_c$	87.1%
$-A_w$	86.4%
$-A,A_c$	85.7%
$-A,A_w$	85.2%
$-A_c,A_w$	84.8%
$-A,A_c,A_w$	63.7%
−GSF extractor	81.3%

length: <10, [10,20), [20,30), [30,40), and neglect of length ≥40 (due to limited training samples). In the Semeval2010Task8, the model presented in this paper performs better than the other models in all five categories. Overall, the model presented in this paper provides better results regardless of whether the input sentence is long or short. This is closely related to the model providing as much information as possible about the key nodes and subtrees of the SDP.

4.7 | Influence of renormalisation parameters

In order to ensure the numerical stability of GCN networks, renormalisation techniques should be introduced. Renormalisation parameter γ is often an important factor affecting the performance of GCN model. It is clear from Figure 3 that the model in this paper is sensitive to γ in the range [0.25,1.75]. When $\gamma = 0.8$ and $\gamma = 1.15$, the model presented in this paper has better performance.

4.8 | Testing on industrial datasets

In order to verify the performance of the proposed algorithm in actual industrial application scenarios, a large scale data set of industrial equipment fault diagnosis and root cause analysis is collected, and good results are obtained. The specific data set is described as follows: There are 27,047 pieces of data in total, and there are four kinds of relationships, namely root cause,

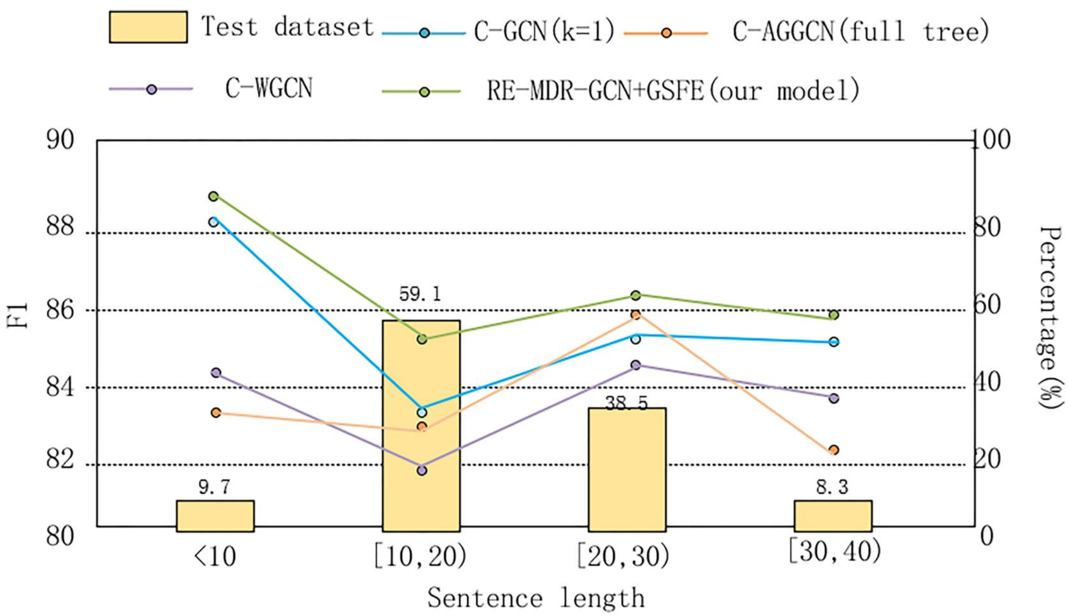

FIGURE 2 F1 values of the model under the Semeval2010Task8 test set for different sentence lengths.

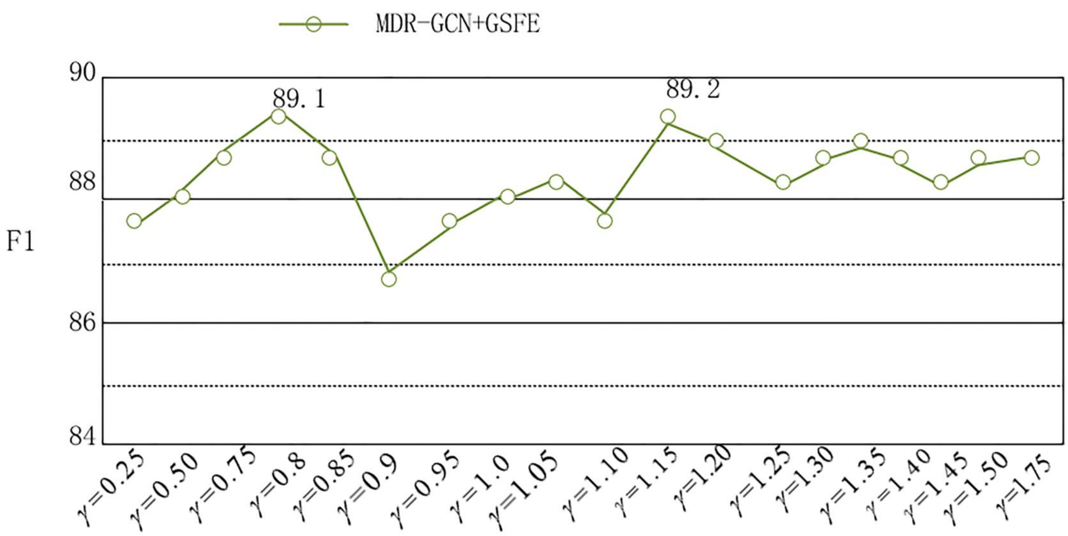

FIGURE 3 Adjust the renormalisation parameter to the F1 value of the model on the Semeval2010Task8 test set.

first step, next step and result. For example, in one of the maintenance records, the problem is described as the fault of the safety door, and the process is described as follows: (1) The manipulator cannot be powered on, and it is found that there is a problem with the latch of the front door of 30B after the inspection; (2) There are no spare parts on site for the time being, and the communication connection of 30B is disconnected for the production schedule; (3) Repair other devices first. (4) Find the safety door switch after the device is stable, and replace the safety module properly. The device also turned on normally. (5) Heat engine for 5 min after starting. Automatic operation without alarm delivery production. Given the sentence entity set, a total of 10 triples can be summarised from the above process description, respectively: (1) The root cause of the safety door fault is the problem of the safety door latch, (2) The first step of the safety door fault is to check the front door 30B, (3) The robot cannot be powered on. Next check the front door safety door 30B, (4) Check the front door safety door 30B, (5) Safety door latch problem next disconnect communication connection, (6) Disconnect communication connection result manipulator recovery, (7) Disconnect the communication connection result The device is stable, (8) Disconnect the communication connection result The security module is normal, (9) Disconnect the communication connection result The safety door is open normally. (10) Disconnect the communication connection result. The heat engine has no alarm. Entity relationship extraction was carried out through the model proposed in this paper, and the average F1 value was 70.2%.

5 | CONCLUSIONS

In this paper, a new model based on GCN is proposed, which can be better used for relational extraction by using multiple forms of dependency tree information. Compared with the hard pruning strategy, the semantic relation extractor proposed in this paper can encode the whole dependency structure more comprehensively and flexibly, and better balance the favourable additional information and the unfavourable interference in the sentence. This article adds a new GSF Extractor module. This module uses the attention mechanism to capture deeper semantic features from the context representation as the global semantic features of the input text, thus helping the model to capture deeper semantic information at the sentence level for relational extraction tasks. The validity of the proposed model is demonstrated by experiments in SemEval2010-Task8 and KBP37 data sets. In the experiment, the importance of renormalisation parameters is also pointed out and it is verified that the model based on GCN is more sensitive to renormalisation parameters. Appropriate adjustments to this parameter are necessary to get the best results in the GCN model. In future work, we should further study the renormalisation process both theoretically and experimentally, and consider how to extend our method to longer input tasks and multi-relation extraction tasks (such as document-level relation extraction).

AUTHOR CONTRIBUTIONS

Zhao Liangfu, Xiong Yujie, Gao Yongbin, Yu Wenjun contributed to the conception of the study; Zhao Liangfu performed the experiment; Zhao Liangfu, Xiong Yujie, Gao Yongbin contributed significantly to analysis and manuscript preparation; Zhao Liangfu performed the data analyses and wrote the manuscript; Gao Yongbin, Yu Wenjun helped perform the analysis with constructive discussions.

ACKNOWLEDGEMENTS

This work was supported in part by the Science and Technology Innovation 2030 - Major Project of 'New Generation Artificial Intelligence' granted by Ministry of Science and Technology (Grant No. 2020AAA0109300), in part by Shanghai Local Capacity Enhancement project (No. 21010501500), in part by 'Science and Technology Innovation Action Plan' of Shanghai Science and Technology Commission for social development project under Grant 21DZ1204900.

CONFLICT OF INTEREST STATEMENT

We declare that we have no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available in roomylee/cnn-relation-extraction at https://github.com/roomylee/cnn-relation-extraction/tree/master/SemEval2010_task8_all_data and zhangdongxu/kbp37 at https://github.com/zhangdongxu/kbp37. These data were derived from the following resources available in the public domain: https://github.com/roomylee/cnn-relation-extraction/tree/master/SemEval2010_task8_all_data and https://github.com/zhangdongxu/kbp37.

ORCID

Zhao Liangfu https://orcid.org/0009-0003-2591-2962

REFERENCES

1. Zhou, J., et al.: Swarm intelligence-based task scheduling for enhancing security for IoT devices. IEEE Trans. Comput. Aided Des. Integrated Circ. Syst. 42(6), 1756–1769 (2023). https://doi.org/10.1109/TCAD.2022.3207328
2. Zheng, S., et al.: Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. Association for Computational Linguistics, Stroudsburg, USA (2017)
3. Zhou, J., et al.: Temperature-constrained reliability optimization of industrial cyber-physical systems using machine learning and feedback control. IEEE Trans. Autom. Sci. Eng. 20(1), 20–31 (2023). https://doi.org/10.1109/TASE.2021.3062408
4. Zhou, J., et al.: Decomposed task scheduling for security-critical mobile cyber-physical systems. IEEE Internet Things J. 9(22), 22280–22290 (2022). https://doi.org/10.1109/JIOT.2021.3081434
5. Zhang, Y., Qi, P., Manning, C.D.: Graph convolution over pruned dependency trees improves relation extraction. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2205–2215. ACL, Stroudsburg (2018)
6. Park, C., Park, J., Park, S.: AGCN: attention-based graph convolutional networks for drug-drug interaction extraction. Expert Syst. Appl. 159, 113538–113550 (2020). https://doi.org/10.1016/j.eswa.2020.113538
7. Li, J., et al.: Index modulation multiple access for 6G communications: principles, applications, and challenges. IEEE Network 37(1), 52–60 (2023). https://doi.org/10.1109/mnet.002.2200433

8. Li, J., et al.: Composite multiple-mode orthogonal frequency division multiplexing with index modulation. IEEE Trans. Wireless Commun. 22(6), 3748–3761 (2023). https://doi.org/10.1109/twc.2022.3220752
9. Kambhatla, N.: Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In: Proceedings of the ACL 2004 on Interactive Poster and Demonstrational Session, p. 22. ACL, Stroudsburg (2004)
10. Bunescu, R.C., Mooney, R.J.: A shortest path dependency kernel for relation extraction. In: Proceedings of the 2005 Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pp. 724–731. ACL, Stroudsburg (2005)
11. Zeng, D., et al.: Relation classification via convolutional deep neural network. In: Proceedings of the 25th International Conference on Computational Linguistics, Dublin, Aug 23-29, 2014, pp. 2335–2344. ACL, Stroudsburg (2014)
12. Zhang, D., Wang, D.: Relation Classification via Recurrent Neural Network. (2015-11-25) [2020-06-16] https://arxiv.org/pdf/1508.01006.pdf
13. Santos, C.N.D., Xiang, B., Zhou, B.: Classifying Relations by Ranking with Convolutional Neural Networks. (2015-05-24) [2020-05-14] http://www.arxiv.org/pdf/1504.06580.pdf
14. Wang, L., et al.: Relation classification via multi-level attention CNNs. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 1298–1307. ACL, Stroudsburg (2016)
15. Zhang, X., Chen, F., Huang, R.: A combination of RNN and CNN for attention-based relation classification. Procedia Comput. Sci. 131, 911–917 (2018). https://doi.org/10.1016/j.procs.2018.04.221
16. Tang, L., et al.: Convolutional LSTM network with hierarchical attention for relation classification in clinical texts. In: Proceedings of the 2019 International Joint Conference on Neural Networks, pp. 1–8. IEEE, Piscataway (2019)
17. Xu, Y., et al.: Classifying relations via long short term memory networks along shortest dependency paths. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1785–1794. ACL, Stroudsburg (2015)
18. Guo, X., et al.: A single attention-based combination of CNN and RNN for relation classification. IEEE Access 7, 12467–12475 (2019). https://doi.org/10.1109/access.2019.2891770
19. Miwa, M., Bansal, M.: End-to-end relation extraction using LSTMs on sequences and tree structures. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 1105–1116. ACL, Stroudsburg (2016)
20. Hu, Y., et al.: A graph convolutional network with multiple dependency representations for relation extraction. IEEE Access 9, 81575–81587 (2021). https://doi.org/10.1109/ACCESS.2021.3086480
21. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: Proceeding of 5th International Conference on Learning Representations, Palaisdes Congres Neptune, Toulon, France, pp. 1–14 (2017)
22. Bruna, J., et al.: Spectral networks and locally connected networks on graphs. In: Proceeding of 2nd 2nd International Conference on Learning Representations, Banff, AB, Canada, pp. 1–14 (2014)
23. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: D.D. Lee, et al., Eds Proceeding of AduNeural Information Processing System, vol. 29, pp. 3844–3852 (2016)
24. Wu, F., et al.: Simplifying graph convolution-al networks. In: Proceeding of 36th International Conferences on Machine Learning, Long Beach, CA, USA, pp. 6861–6871 (2019)
25. Shuman, D.I., et al.: The emerging field of signal processing on graphs: extending highdimensional data analysis to networks and other irregular domains. IEEE Signal Process. Mag. 30(3), 83–98 (2013). https://doi.org/10.1109/msp.2012.2235192
26. Zhou, P., et al.: Attention-based bidirectional long short-term memory network for relation classification. In: Proceeding of 54th Annual Meeting of the Association for Computational Linguistics, pp. 207–212. ACL, Stroudsburg (2016)
27. Lee, J., Sangw, S., Suk, C.: Semantic relation classification via bidirectional LSTM networks with entity-aware attention using latent entity typing. Symmetry 11(6), 785 (2019). https://doi.org/10.3390/sym11060785
28. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. In: Annual Meeting of the Association for Computational Linguistics, 7th International Joint Conferences on Natural Language Processing, Beijing, China, vol. 1, pp. 1556–1566 (2015)
29. Zhang, Y., et al.: Position-aware attention and supervised data improve slot fifilling. In: Proceedings of the Conference on Empirical Methods Natural Language Processing, Copenhagen, Denmark, pp. 35–45 (2017)
30. Guo, Z., Zhang, Y., Lu, W.: Attention guided graph convolutional networks for relation extraction. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp. 241–251 (2019)

How to cite this article: Liangfu, Z., et al.: Multiple dependence representation of attention graph convolutional network relation extraction model. IET Cyber-Phys. Syst., Theory Appl. 9(3), 247–257 (2024). https://doi.org/10.1049/cps2.12080